Add the following code to your python code:

|  |
| --- |
| employee\_dict = {'Jimmy': 25.00, 'Tom': 48.50, 'Alice': 6.00}  meal\_dict = {'BBM': [7.10, 'Big Breakfast Meal'],  'TBE': [9.90, 'Turkey Bacon with Egg' ],  'MSO': [7.50, 'Mushroom Swiss Omelette'],  'NLC': [6.60, 'Nasi Lemak with Chicken'],  'CBV': [11.90, 'Curry Beef with Vegetable']}  purchase\_list=[] |

* **employee\_dict** contains the employee names and their card balance. For example, employee Alice has a card balance of $6.00
* **meal\_dict**stores the meal information, including meal code, price and description.
* **purchase\_list** stores the list of  the meal descriptions that have been purchased by the employee.

You are to implement a program for employees to purchase the meal from company canteen using their pre-paid cash cards. Write a function called **purchase\_meal** that:

* takes in an employee name as a parameter.
  + If the employee's name is not found in employee\_dict, display error message and function ends.
  + If employee name is correct, display message with card balance
* prompts user to enter a meal code and check the validity of the code. If the code is not correct, it will keep asking user to enter the code until the code is correct
* allows user purchase multiple meals if there are sufficient balance.
* stores the item purchased in the purchase\_list given
* updates the balance in the employee\_dict and display the card balance after the purchase
* prints the purchase meals before function ends.

The expected program output is as follows:

|  |  |
| --- | --- |
| Test Code | Expected Output |
| purchase\_meal('Mary') | Invalid employee name. |
| purchase\_meal('Jimmy') | You have the balance of $25.0 Would you like to purchase meal? (Y/N): Y Enter meal code: ABC Meal code is invalid! Enter meal code: BCD  Meal code is invalid! Enter meal code: TBE  Turkey Bacon with Egg redeem successful  You have the balance of $15.1 Would you like to purchase meal? (Y/N): Y Enter meal code: MSO Mushroom Swiss Omelette redeem successful You have the balance of $7.6 Would you like to purchase meal? (Y/N): N Meals purchased: Turkey Bacon with Egg, Mushroom Swiss Omelette |
| purchase\_meal('Alice') | You have the balance of $6.0 Would you like to purchase meal? (Y/N): Y Enter meal code: BBM Big Breakfast Meal purchase unsuccessful, insufficient balance for purchasing. You have the balance of $ 6.0 Meals purchased: NULL |

Question 5 options:

|  |
| --- |
|  |

Add the following code to your python code:

|  |
| --- |
| inventory\_info = {  '1': {  'Product Name': 'Apple',  'prices': [1.2,1.5,1.3]},  '2': {  'Product Name': 'Banana',  'prices': [0.8, 0.9, 0.7, 0.6, 0.6]},  '3': {  'Product Name': 'Orange',  'prices': [1.0, 1.2, 0.9, 1.1]},  } |

Write a function prod\_info() to display the product item details. You must **use for or while loop**to get each product information. Your program should be able to handle a varying number of items in the dictionary, not limited to the three items given in the above code.

The expected program output is as follows:

|  |
| --- |
| Product Information: 3 P1.Apple      Highest Price: 1.5      Average Price: 1.33  P2.Banana      Highest Price: 0.9      Average Price: 0.72  P3.Orange      Highest Price: 1.2      Average Price: 1.05 |

Question 4 options:

|  |
| --- |
|  |

The score information is stored in a dictionary, subject\_dict, where the subject name is key, and the score of each subject is the value.

 Add the following code to your python code.

|  |
| --- |
| subject\_dict = {  "UX Design": 80,  "Programming": 70,  "Data Modelling": 80,  "Network Administration": 60 } |

Write a program to display the subject(s) with the highest score, showing the subjects in alphabetical order if there is more than one subject.

Your program should be able to handle a varying number of items in the dictionary, not limited to the four items given in the above code.

The expected program output is as follows:

|  |
| --- |
| Subject(s) with highest score:  ['Data Modelling', 'UX Design'] |

Question 3 options:

|  |
| --- |
|  |

**Question 2**(25 points)

You are to create a program to split bills with friends. The program captures name and price of purchased items. After the user entered all the items, the program will  
•    Display the items that have purchased  
•    Display the total amount including additional 10% as the transportation fee  
•    Get the number of friends who wish to share the amount  
•    Compute the amount to be borne by each person and display it. The amount is to be evenly distributed all friends.

The expected program output is as follows:

|  |
| --- |
| ==== Contribution calculator ===  Enter 1 to add the name of purchased item Enter 0 to end the input of purchased item Please select: **1** Enter the name of the purchased item: **Birthday Cake** The price: $**30**  Enter 1 to add the name of purchased item Enter 0 to end the input of purchased item Please select: **1** Enter the name of the purchased item: **Party Room Decoration** The price: $**10**  Enter 1 to add the name of purchased item Enter 0 to end the input of purchased item Please select: **0**  Please enter the number of pax to share the amount: **4**  Purchased items:  {Birthday Cake, Party Room Decoration} Total amount spent (including 10% additions for transportation):$44.00 Each friend to pay:$11.00 |

Question 2 options:

|  |
| --- |
|  |

Write a problem to capture student's score for **4** subjects. The program checks that the score is a number ranging from 0 to 100 (inclusive) before adding it to a list. You can assume that user will always enter a number. You must use **for or while loop** in your program.

The expected program output is as follows:

|  |
| --- |
| Enter the score for subject 1: **70** Enter the score for subject 2: **100** Enter the score for subject 3: **-45** A score must be between a range of 0 and 100! Enter the score for subject 3: **-78** A score must be between a range of 0 and 100! Enter the score for subject 3: **89** Enter the score for subject 4: **87**  Score list in descending order is: [100, 89, 87, 70] |

Question 1 options:

|  |
| --- |
|  |